

## BÀI KIỂM TRA THỰC HÀNH

### MÔN HỆ ĐIỀU HÀNH

**THỜI GIAN: 7h – 11h30, ngày 8/5/2022.**

Sinh viên sau khi thực hiện xong, lưu tất cả file mà mình đã thực hiện (bao gồm file .c, .out) vào 1 thư mục, đồng thời tạo ra 1 file có tên là lienhe.txt, trong đó ghi rõ họ tên, số điện thoại, địa chỉ email để giáo viên có thể liên hệ khi gặp vấn đề về file, lưu file này vào thư mục vừa tạo. Nén thư mục thành file .rar. Đặt tên file .rar có dạng: Mã sinh viên\_họ tên.rar.

Ví dụ: sinh viên Nguyễn Văn An, có mã sinh viên: 123. Đặt tên file: 123\_NguyenVanAn.rar

Lưu ý: sinh viên có thể tham khảo mọi tài liệu mình có, nhưng không được sao chép bài của bạn.

Sinh viên lên link sau để download các file hỗ trợ thực hiện kiểm tra.

[https://drive.google.com/drive/folders/1fiHbGaHl4GFItSMdncX1UfBcE89A35cF?usp=s\\_haring](https://drive.google.com/drive/folders/1fiHbGaHl4GFItSMdncX1UfBcE89A35cF?usp=s_haring)

Các file dữ liệu (Ben, Alex) được đặt trùng với tên của người đấu giá, các giá trị trong file là số tiền có thể đặt trong mỗi lần đấu giá (mặc định số tiền sẽ được đặt tăng dần, do không có ai đấu giá mà lần sau đặt thấp hơn lần trước).

Với thư viện, các biến toàn cục và các cấu trúc đã được chuẩn bị sẵn, sinh viên thực hiện theo các yêu cầu sau:

#### **Phần căn bản (5đ)**

Sinh viên thực hiện xây dựng hàm **void \*xuly (void \*Dulieu)** sao cho khi thực thi, hàm này sẽ mở file được lưu trữ ở biến **filename[BUFFER\_SIZE]** trong tham biến được truyền vào thuộc cấu trúc data, sau đó đọc toàn bộ dữ liệu ở file lên mảng **value[BUFFER\_SIZE]**, lưu lại số phần tử của mảng ở biến **length** trong file tham biến được truyền vào thuộc cấu trúc data. **(2đ)**

Tại hàm **main()** sinh viên khai báo 1 biến thuộc cấu trúc data, cho truyền đối số **argv** là tên file cần xử lý vào biến **filename[BUFFER\_SIZE]** trong cấu trúc data. **(1đ)**

Sau đó tạo ra 1 tiểu trình, với đối số truyền vào là biến thuộc cấu trúc data vừa khai báo. Tiểu trình này thực hiện gọi đến hàm **xuly** để thực hiện tính toán. **(1đ)**

Sau khi chờ tiểu trình hoàn thành, hãy xuất ra thông tin của biến thuộc cấu trúc data đã xử lý ở trên. Thông tin sẽ xuất có dạng như sau: **(1đ)**

*Người đấu giá là: < xuất biến filename>*

*Dữ liệu: < xuất ra toàn bộ giá trị của biến value>*

**Phần nâng cao: gồm 2 option:**

**Option 1: độ phức tạp xử lý không cao – điểm số đạt được: 7đ**

**Option 2: độ phức tạp xử lý cao – điểm số đạt được: 10đ**

**Option1:**

Sau khi đã hoàn thành phần căn bản, hãy tạo ra **message** để gửi dữ liệu là biến thuộc cấu trúc data đã xử lý ở trên. Sử dụng biến **mesg\_type** đã được chuẩn bị sẵn để thực hiện truyền dữ liệu. Hãy gửi biến chứa các dữ liệu đã xử lý sang cho tiến trình ToChuc.

Dựa vào file ToChuc.c đã cung cấp, hãy viết chương trình nhận dữ liệu từ tiến trình DauGia. Sau đó xuất ra toàn bộ thông tin đã nhận được. Thông tin xuất ra có dạng: **(2đ)**

*Người đấu giá là: < xuất biến filename>*

*Dữ liệu: < xuất ra toàn bộ giá trị của biến value>*

**Option2:**

Sau khi đã hoàn thành phần căn bản, hãy xử lý việc tạo ra tiến trình sao cho có thể tạo ra 2 hay nhiều tiến trình cùng lúc, mỗi tiến trình sẽ xử lý 1 file được truyền từ đối số.

Hãy cho chạy lại các tiến trình này để lấy dữ liệu.

Hãy thực hiện hàm **void \*daugia (void\* Dulieu)** đã được chuẩn bị ở file **daugia.c**, hàm này khi thực hiện sẽ làm công việc sau: **(2đ)**

Giả sử rằng có nhiều tập tin mà mỗi tập tin (đại diện cho một người), dữ liệu của tập tin này đã được đưa vào biến thuộc cấu trúc data, đã xử lý ở phần trên. Biến **value[BUFER\_SIZE]** chứa các số tiền của mỗi lần đặt lệnh. Chương trình cần tạo ra nhiều tiến trình, mỗi tiến trình sẽ đọc một tập tin và các tiến trình đồng thời đặt lệnh đấu giá, có các tình huống sau có thể xảy ra:

- Giá đấu (biến **value[BUFER\_SIZE]**) cao hơn giá hiện tại (biến **result**) (giá hiện tại là biến số chia sẻ và khởi tạo là 0): giá hiện tại sẽ được cập nhật và tên người thắng (biến **name**) cũng sẽ được cập nhật (là tên của tập tin chứa giá đấu đang được xử lý đang được lưu trữ trong biến **filename**).

- Giá đấu thấp hơn hay bằng giá hiện tại: không cập nhật gì cả.

- Giá đấu cao hơn giá hiện tại nhưng nếu giá hiện tại cao nhất cũng là của người đấu này thì tiến trình này bị chặn lại cho đến khi người thắng được cập nhật. (Không ai tự bỏ giá cao hơn giá vừa bỏ liên trước đó).

Sau khi đã xử lý xong hàm **void \*daugia (void\* Dulieu)** hãy tạo ra số tiểu trình tương ứng với số file để các tiểu trình thực hiện đấu giá với nhau. **(1đ)**

Sau khi tất cả tiểu trình đã hoàn thành, hãy thực hiện khai báo 1 biến thuộc cấu trúc **message** lưu tất cả giá trị của người thắng (biến **name**) vào biến **won[BUFER\_SIZE]**, số tiền thắng biến **result** vào biến **money**. Thực hiện gửi toàn bộ thông tin của biến **message** này sang cho tiến trình ToChuc. **(1đ)**

Dựa vào file ToChuc.c đã cung cấp, hãy viết chương trình nhận dữ liệu từ tiến trình DauGia. Sau đó xuất ra toàn bộ thông tin đã nhận được. Thông tin xuất ra có dạng: **(1đ)**

*Ngươi chiến thắng là: < xuất biến won>*

*Số tiền: < xuất ra toàn bộ giá trị của biến money>*

**HẾT**